Replace the paragraph beginning at page 3, line 11 with:

Therefore, the programmable controller may use a configuration for receiving the instruction list and executing sequential processing in accordance with instructions shown by the instruction list. In this case, the programmable controller sequentially executes a control program by interpreting the received instruction list line by line, that is, by converting the list into code directly executable by a microprocessor mounted on the programmable controller. The programmable controller is hereafter referred to as interpreter-type programmable controller.

Replace the paragraph beginning at page 3, line 22 with:

Moreover, a compiling-type programmable controller is practically used as another type of programmable controller that receives directly-executable code from a control-program-development supporting apparatus through a microprocessor in the programmable controller and executes the codes. In the case of the configuration using the compiling-type programmable controller, a control-program-development supporting apparatus is provided with a compiler, which converts a ladder diagram directly or temporarily into an instruction list by the compiler and then compiles the list into code directly executable by the microprocessor and transfers the execution codes to the programmable controller.

Replace the paragraph beginning at page 4, line 20 with:

However, because the above control program depends on the vendor or type of programmable controller, it is impossible to execute the same sequential processing between different types of programmable controllers by using control programs having the same source code. Therefore, it is necessary to develop a new control program or transplant a new model or different model to a programmable controller and thus, it is difficult to effectively use past properties of a control program.

Replace the paragraph beginning at page 5, line 1 with:

For example, according to the advanced-language sequence instruction program generator and universal sequencer disclosed in Japanese Patent Laid-Open No. HEI 7-

295612, the advanced-language sequence instruction program generator translates a language sequence instruction program into an advanced-language sequence instruction program with a translation program and compiles the advanced-language sequence instruction program to generate an execution-type sequence instruction program. Or, the universal sequencer executes the advanced-language sequence instruction program while sequentially interpreting the program. Thereby, it is possible to execute sequential control with the same control program independently of the vendor or type of a sequencer.

Replace the paragraph beginning at page 6, line 24 with:

Firstly, there is a problem that, because a microprocessor in a universal computer (hereafter referred to as universal Microprocessor), such as a personal computer, operates with an advanced function at a high speed and has become inexpensively available in recent years, the advantage of developing a microprocessor in a conventional programmable controller with an ASIC (Application Specific IC) is slowly decreasing.

Replace the paragraph beginning at page 7, line 7 with:

Particularly, a universal microprocessor is provided with primary and secondary caches and constructed in accordance with acceleration of processing techniques such as pipeline processing, super-scaler processing and out-of-order processing and moreover, execution codes directly executable by the universal microprocessor are also generated by a compiler for converting the execution codes into execution codes forming an optimum code system by fully using the acceleration techniques.

Replace the paragraph beginning at page 8, line 14 with:

Fourthly, though the "advanced-language sequence instruction program generator" disclosed in the above Japanese Patent Laid-Open No. HEI 7-295612 generates an object file by converting a source file described with the control program of an instruction list or the like into the source file of an advanced programming language such as C language, if only the source file of the advanced programming language is corrected. There is a problem that two source files showing the same sequential processing do not match each

A8 cont

other because the source file of an original control program is not changed due to the above correction.

Replace the paragraph beginning at page 8, line 25 with:

A9

Particularly, in the case of a control-program-development supporting apparatus capable of developing a control program with an advanced programming language, to perform step execution by using a debugging tool, for example, step execution is possible for one line of the advanced programming language but it is impossible to perform step execution every line for a language configuration such as an instruction list before conversion into an advanced programming language.

Replace the paragraph beginning at page 9, line 23 with:

A10

Fifthly, to realize the above-described under-RUN writing method, there is a problem that a memory having the same size as a memory in which a control program currently executed is read must be separately prepared. Because a memory is generally expensive among hardware parts, a maker or user must make a large investment in order to realize the under-RUN writing method.

Replace the paragraph beginning at page 11, line 20 with:

A11

An instruction interpreting unit is generally described with a structure of comparing input data with a pattern of the interpretation side and when the data and the pattern match each other, executing a previously-stored operation (action). Therefore, pattern matching of the instruction interpreting unit requires a lot of time when the operation is executed in the case of the interpreter type and when the operation is compiled in the case of the compiling type. Thus, both types have factors impeding productivity.

Replace the paragraph beginning at page 24, line 12 with:

A12

Fig. 1 is a block diagram showing a schematic configuration of a programmable controller of a first embodiment;

Replace the paragraph beginning at page 24, line 18 with:

Fig. 3 is an illustration for explaining the generation of an execution code followed by division of a control program in a control-program-development supporting apparatus of a second embodiment;

Replace the paragraph beginning at page 24, line 23 with:

Fig. 5 is a table showing the relation between the number of steps and a divided file in the control-program-development supporting apparatus of the second embodiment;

Replace the paragraph beginning at page 25, line 20 with:

Fig. 11 is an illustration for explaining the generation of an execution code followed by division of a control program and conversion of the program into an advanced programming language in a control-program-development supporting apparatus of a third embodiment;

Replace the paragraph beginning at page 26, line 17 with:

Fig. 15 is an illustration for explaining the generation of an execution code of a control-program-development supporting apparatus of a fourth embodiment;

Replace the paragraph beginning at page 26, line 21 with:

Fig. 16 is a table showing sample programs and their sizes, and processing times in a control-program-development supporting apparatus of a fifth embodiment;

Replace the paragraph beginning at page 26, line 25 with:

Fig. 17 is an illustration for explaining the development environment and execution environment of a control program in a control-program-development supporting apparatus of a sixth embodiment;

Replace the paragraph beginning at page 27, line 16 with:

Fig. 22A and Fig. 22B are illustrations for explaining the replacement of an execution code using a binary patch in a programmable controller of a seventh embodiment;

Replace the paragraph beginning at page 27, line 19 with:

Fig. 23 is an illustration for explaining the generation of an execution code in a control-program-development supporting apparatus of an eighth embodiment;

Replace the paragraph beginning at page 27, line 23 with:

Fig. 24 is an illustration showing a source file that is the content of a control program in the control-program-development supporting apparatus of the eighth embodiment;

Replace the paragraph beginning at page 28, line 2 with:

Fig. 24 is an illustration showing a source file that is the content of a control program in the control-program-development supporting apparatus of the eighth embodiment;

Replace the paragraph beginning at page 28, line 2 with:

Fig. 25 is an illustration showing a C-language expression obtained by converting a compressed file in the control-program-development supporting apparatus of the eighth embodiment;

Replace the paragraph beginning at page 28, line 6 with:

Fig. 26 is an illustration for explaining compiling in a control-program-development supporting apparatus of a ninth embodiment; and

IN THE CLAIMS:

Replace the indicated claims with: